

REMARKS

The claims are 4 and 8-21.

Applicant presents herewith corrected Figures 1-4. The numeral 11 has been stricken from various steps of the process shown in Figures 1-3. The reference sign 25 and its element (box) have been removed from Figure 4. Accordingly, it is submitted, the drawings are now in condition for allowance.

Applicant has also deleted claims 6 and 7 and amended claim 9. It is submitted that the amendment to claim 9 obviates the Examiner's rejection under 35 USC 112 to that claim and to its dependant claim 10.

Applicant had requested an Interview along with a suggested amendment. The Examiner did not grant the Interview, and so the suggested amendment was not entered.

Upon further review of the prior art and the claims in suit, Applicant does not believe the amendment is necessary for patentability over the principal reference, U.S. Patent No. 6,151,708 to Pedrizetti, et. al.

In the 35 U.S.C. § 102(e) rejection to claim 4, the Examiner states that Pedrizetti's col. 1, lines 48-59 "shows that the hash table and the update data information is transferred to the client for processing and is compared with the client to determine whether or not the actual update data should be transferred as well." But, with all due respect, col. 1, lines 48-59 does not show the limitations of claim 4 – that a hash is transferred from server to client. In fact, Pedrizetti nowhere transmits a hash of a file. Therefore, Applicant respectfully traverses the Examiner's rejection to claim 4.

The apparatus and process of Pedrizetti compares programs on the client with

programs on the server to determine whether updates are necessary. (Col. 1, lines 41-45.)

It compares those programs not by transmitting hashes, but by transmitting a bitmap table indicating the presence of a hash value. That hash value may or may not refer to a specific programs to be updated – Pedrizetti acknowledges that its method may lead to “hash collisions.” In order to resolve “hash collisions” Pedrizetti then transfers an index data file. Finally, an actual updated programs may finally be transferred.

Where confusion may have arisen between claim 4 of the present invention and Pedrizetti is in Pedrizetti’s use of a “hash function.” A hash function (or other “one to many mapping function”) is used on program “identifiers” to determine if updating is necessary, by converting those program identifiers into a hash value, and then populating a bitmap sparse table according to the hash table values. (Col. 1, lines 46-48, Col. 4, lines 50-65, Figure 4.)

However, it is not the hash values in Pedrizetti that are sent from the server to the client – it is the bitmap sparse table that is sent from a server to a client, where it is compared to a client side bitmap table to determine if a bit is present or not for any particular program identifier. (Col. 4, lines 50-65, Col. 5 lines 7-20.)

Thus, for example, as shown by Figure 4 of Pedrizetti, a bit may be set high, indicating a hash value of 8443 is present on the server. The presence of a particular hash value on the server, through a bit being set high on the bitmap table, means that that programs associated with that particular hash value are present on the server. If the client has that bit also set high, the hash value exists on the client. If the hash value exists on the client, one or more programs exist on the client with that hash value.

But the presence of the hash value in the bitmap table of the client in Pedrizetti

only indicates that one or more programs exist with the hash value on the client, that is, the presence of any particular hash value on the client or server through a bit set high on the bit map does not indicate a specific program is present on both server and client and needs updating on the client. Rather, the presence of a particular hash means only that more than one program with that hash value may exist –a “hash collision.” Pedrizetti must then resolve that hash collision. In order to resolve the hash collision, and make sure the program present on the server is the program desired by the client, Pedrizetti then transfers an “data (index) file” for that hash value. (Col. 5, lines 47-50, Col. 7, lines 22-24.) This data index file is comprised of file names and other identifier information. This second transfer is then compared with the data index file on the client machine to make sure this information matches. (Col. 5, lines 65 – col. 6, line 1) (Interestingly enough, apparently even this information may still not be unique and lead to further collisions, (Col. 7, l. 56-65) although Pedrizetti appears to offer no solution to this difficulty.)

Pedrizetti’s transmission of bit table rather than a hash is because, Pedrizetti claims:

Since the client's update evaluation potentially requires checking a huge number of files for available updates, it would be prohibitive to have the client and server pass file name (or other program or hardware module references) back and forth to determine the availability of upgrades. Instead, as discussed below, the server maintains a large bit field having bit entries which indicate the potential availability of updates. This bit field is compressed and transferred to the client, allowing the client to locally determine a correspondence between the client's list and the server's bit field. The correspondence between modules and upgrades is by a hash function which maps unique module references to index positions within the field (a hash table). (Col. 4, lines 47-58.)

Thus, Pedrizetti functions differently from the invention of claim 4, which reads:
“...said means for obtaining data information from said distribution media obtains said hash from said means for updating present on said distribution media, *which hash is transmitted through said means for transmission to said means for processing, and which upon receipt of said hash of said data information compares said hash with said target in order to determine if said data should be transmitted to said target.*”

Pedrizetti neither transmits a hash, nor compares a hash with a target.

Thus, it is submitted that the limitations of claim 4 are not met by the Pedrizetti reference, and it is respectfully requested that the Examiner's rejection be withdrawn and the claim proceed to issue.

Claim 8 is similar to claim 4 and Applicant respectfully traverses the Examiner's rejection, under 35 U.S.C. 102 §(e) as being anticipated by Pedrizetti to claim 8.

As Applicant noted above, Pedrizetti fails to teach, suggest or disclose transmitting a hash, unlike the limitations of claim 8: “...transmitting a hash of data information from a first distribution media to said target; [and] comparing said hash in order to determine if data information should be transmitted to said target...” Thus, it is submitted, Pedrizetti cannot serve as anticipatory reference to claim 8, and it is respectfully requested that the Examiner's rejection be withdrawn and the claim proceed to issue.

Claims 9-20 depend, directly or indirectly, from claim 8 and therefore share the limitations of that claim. Applicant respectfully traverses the Examiner's rejection under 35 U.S.C. 102 §(e) as being anticipated by Pedrizetti to claims 9-20. As was noted above, Pedrizetti nowhere, teaches, suggests nor discloses the limitations of claim 8, and so

cannot be held to disclose the limitations of claims 9-20. Thus, it is submitted, Pedrizetti cannot serve as anticipatory reference to claims 9-20, and it is respectfully requested that the Examiner's rejection be withdrawn and the claim proceed to issue.

Applicant respectfully traverses the Examiner's rejection under 35 U.S.C. 102 §(e) as being anticipated by Pedrizetti, to claim 21. Claim 21 comprises the limitations of: "...transmitting said hash of data information to said target; [and] comparing said hash in order to determine if data information should be transmitted to said target..." As was noted above with regard to the Pedrizetti reference, it nowhere teaches, suggests or discloses transmitting a hash to a target. Thus, it is submitted, Pedrizetti cannot serve as anticipatory reference to claim 21, and it is respectfully requested that the Examiner's rejection be withdrawn and the claim proceed to issue.

Applicant further traverses the Examiner's rejections to claims 16 and 17 under 35 U.S.C. 102 §(b) as being anticipated by Aviani, U.S. Patent No. 5,950,205. Applicant notes that Aviani nowhere teaches, suggests nor discloses the limitations of independent claim 8 and therefore cannot, it is submitted serve as an anticipatory reference for dependant claims 16 and 17.

Aviani discloses a cache file system. Disk drives within the system may store either meta data or hashes. But there is no teaching, suggestion nor disclosure of transmitting any hashes or data information, as in claim 8. Thus, it is submitted, Aviani cannot serve as anticipatory reference to claims 16 and 17, and it is respectfully requested that the Examiner's rejection be withdrawn and the claim proceed to issue.

Applicant notes the other references cited by the Examiner.

Conclusion

Claims 4 and 8-21 define patentable subject matter over the art of record and are not anticipated by nor obvious in view of the references of record. A Notice of Allowance is respectfully solicited.

Respectfully Submitted,



Joseph E. Chovanes, Esq.
Registration No. 33,481
Attorney for Applicant
Tel.: 610-648-3994
Fax: 610-648-3997

Dated: 2-6-04